



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/867,650	05/31/2001	Michael Anthony Sijacic	06502.0338-00000	7983

22852 7590 01/17/2006

FINNEGAN, HENDERSON, FARABOW, GARRETT & DUNNER
LLP
901 NEW YORK AVENUE, NW
WASHINGTON, DC 20001-4413

EXAMINER

BULLOCK JR, LEWIS ALEXANDER

ART UNIT	PAPER NUMBER
----------	--------------

2195

DATE MAILED: 01/17/2006

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary

Application No.

09/867,650

Applicant(s)

SIJACIC ET AL.

Examiner

Lewis A. Bullock, Jr.

Art Unit

2195

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --
Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

- 1) ☒ Responsive to communication(s) filed on 07 November 2005.
2a) ☐ This action is FINAL. 2b) ☒ This action is non-final.
3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

- 4) ☒ Claim(s) 1-91 is/are pending in the application.
4a) Of the above claim(s) _____ is/are withdrawn from consideration.
5) ☐ Claim(s) _____ is/are allowed.
6) ☒ Claim(s) 1-91 is/are rejected.
7) ☐ Claim(s) _____ is/are objected to.
8) ☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

- 9) ☐ The specification is objected to by the Examiner.
10) ☒ The drawing(s) filed on 27 August 2003 is/are: a) ☒ accepted or b) ☐ objected to by the Examiner.
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
a) ☐ All b) ☐ Some * c) ☐ None of:
1. ☐ Certified copies of the priority documents have been received.
2. ☐ Certified copies of the priority documents have been received in Application No. _____.
3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

- 1) ☒ Notice of References Cited (PTO-892)
2) ☒ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08)
Paper No(s)/Mail Date _____

- 4) ☐ Interview Summary (PTO-413)
Paper No(s)/Mail Date. _____
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: _____

Art Unit: 2195

DETAILED ACTION

Claim Rejections - 35 USC § 101

1. 35 U.S.C. 101 reads as follows:

Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title.

Claims 27-36 are rejected under 35 U.S.C. 101 because the claimed invention is directed to non-statutory subject matter. The cited claims all detail a memory storing a structure of particular data. M.P.E.P. 2106 details

Where certain types of descriptive material, such as music, literature, art, photographs and mere arrangements or compilations of facts or data, are merely stored so as to be read or outputted by a computer without creating any functional interrelationship, either as part of the stored data or as part of the computing processes performed by the computer, then such descriptive material alone does not impart functionality either to the data as so structured, or to the computer. Such "descriptive material" is not a process, machine, manufacture or composition of matter. (Data consists of facts, which become information when they are seen in context and convey meaning to people. Computers process data without any understanding of what that data represents. Computer Dictionary 210 (Microsoft Press, 2d ed. 1994).)

The policy that precludes the patenting of nonfunctional descriptive material would be easily frustrated if the same descriptive material could be patented when claimed as an article of manufacture. For example, music is commonly sold to consumers in the format of a compact disc. In such cases, the known compact disc acts as nothing more than a carrier for nonfunctional descriptive material. The purely nonfunctional descriptive material cannot alone provide the practical application for the manufacture.

The cited claims all detail an arrangement of data and therefore are not statutory.

Claim Rejections - 35 USC § 102

2. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

A person shall be entitled to a patent unless –

(e) the invention was described in (1) an application for patent, published under section 122(b), by another filed in the United States before the invention by the applicant for patent or (2) a patent granted on an application for patent by another filed in the United States before the invention by the applicant for patent, except that an international application filed under the treaty defined in section 351(a) shall have the effects for purposes of this subsection of an application filed in the United States only if the international application designated the United States and was published under Article 21(2) of such treaty in the English language.

3. Claims 1-91 are rejected under 35 U.S.C. 102(e) as being anticipated by CHONG (U.S. Patent Application Publication 2002/0184610).

As to claim 1, CHONG teaches a method for creating an activity (component / object / workflow step) within a process management system (development system / workflow environment), comprising: receiving first data reflecting a class file (class); receiving second data reflecting a data representation file (imported media / variables); packaging the first and second data (via receiving the classes and variable information packaged in a Jar file) (via componentization) (pg. 29, paragraph 431-434); and associating the packaged data with an activity (via all information are stored in a Jar file) that may be used in an automated workflow process (workflow) (via dragging and dropping the icon representing the component to a diagram of the workflow) (pg. 13, paragraph 257 – 258; pg. 14, paragraph 262) to access information external to the process management system (via using the adapters to access the external data sources and manipulate the external data sources) (pg. 23, paragraph 350 – pg. 24, paragraph 352; pg. 24, paragraph 353-354).

As to claim 12, CHONG teaches a method for implementing a custom activity (component / object / workflow step) within a process management environment (development system / workflow environment), comprising: defining a file associated with a custom activity (via the user creating a component and the class of the component via componentization) (pg. 29, paragraph 430-434); assigning a visual representation associated with the custom activity (via packaging the visual icon via componentization) (pg. 29, paragraph 430-434); receiving an indication (via drag and drop operation) reflecting implementation of the custom activity in a workflow process based on a position of the visual representation in a process map (workflow diagram) representing the workflow process (via dragging and dropping the icon representing the component to a diagram of the workflow and linking the instantiated component) (pg. 13, paragraph 257 – 258; pg. 14, paragraph 262); and invoking the file (via sending and loading the file and allowing the component to be used in an application) (pg. 30, paragraph 433).

As to claim 14, CHONG teaches a method for creating and defining a custom activity (component / object / workflow step) within a process management system (development system / workflow environment), comprising: creating at least one file defining properties (various attributes of the component, i.e. items i-viii) associated with the custom activity (pg. 29, paragraph 430-434); and defining a model (visual icon) associated with the custom activity (via packaging the visual icon via componentization)

(pg. 29, paragraph 430-434), wherein the custom activity may be used to access information external to the process manager system (via using the adapters of the component to access the external data sources and manipulate the external data sources) (pg. 23, paragraph 350 – pg. 24, paragraph 352; pg. 24, paragraph 353-354).

As to claim 20, CHONG teaches a method for implementing a custom activity (component / object / workflow step) in a process management system (development system / workflow environment), comprising: creating a process map (workflow diagram) reflecting an automated workflow process (pg. 13-14, paragraph 257-259); creating an image (visual icon) reflecting a custom activity (via packaging the visual icon with the component such that when the component is loaded the icon is displayed) (pg. 29, paragraph 430-434; pg. 14, paragraph 258); and invoking a class (class) defining the custom activity based on a manipulation of the image by a user (via the drag and drop operation) such that the image is placed in the process map (via drag and dropping the component into the workflow diagram) (pg. 13-14, paragraph 257-259), wherein the custom activity exchanges data with resources external to the process management system (via using the adapters of the component to access the external data sources and manipulate the external data sources) (pg. 23, paragraph 350 – pg. 24, paragraph 352; pg. 24, paragraph 353-354).

As to claim 21, CHONG teaches a method for creating a custom activity (component / object / workflow step) in a process management system (development

system / object / workflow step), the custom activity exchanging information with resources external to the process management system (via using the adapters of the component to access the external data sources and manipulate the external data sources) (pg. 23, paragraph 350 – pg. 24, paragraph 352; pg. 24, paragraph 353-354), comprising: receiving a first file and a second file; and archiving the files in an archive file (via receiving the classes and variable information packaged in a Jar file) (via componentization) (pg. 29, paragraph 431-434) such that when the custom activity is activated the archived files are accessed and executed (via sending and loading the Jar file and allowing the component to be used when dragged into the workflow diagram in an application) (pg. 30, paragraph 433).

As to claim 27, CHONG teaches a memory for storing data for access by a process (application) being executed by a processor, the memory comprising: a structure defining a class file (class) and a data representation file (imported media / variables), packaging the files (via receiving the classes and variable information packaged in a Jar file) (via componentization) (pg. 29, paragraph 431-434), assigning an icon (visual icon) representing the packaged files (component) (via the visual icon is part of the Jar file) (pg. 29, paragraph 431-434), and associating the icon with an activity that performs processes defined by the class and data representation files (via drag and dropping the component into the workflow diagram by manipulating the visual icon) (pg. 13-14, paragraph 257-259).

As to claim 30, CHONG teaches a memory for storing data for access by a process (application) being executed by a processor, the memory comprising: a structure for maintaining an identity of a custom activity (component / object / workflow step), parameters associated with the custom activity (parameters), a first hashtable reflecting data values to be used as input argument in a method, and a second hashtable reflecting output arguments of the method (via the functional black-box interface) (pg. 29, paragraph 430-434).

As to claim 31, CHONG teaches a memory for storing data for access by a process (application) being executed by a processor, the memory comprising: a structure for defining a value of a parameter associated with an input hashtable (variable to be passed in via the black-box interface / data bindings) (pg. 29, paragraph 430-434; pg. 24, paragraph 353-357), mapping a value of a parameter associated with an output hashtable (variable to be passed out via the blackbox interface / data bindings) (pg. 29, paragraph 430-434; pg. 24, paragraph 353-357), and defining a user interface associated with a custom activity (visual icon representing the component) (pg. 29, paragraph 430-434) that performs a process based on the values of the parameters in the input and output hashtables (via using the adapters to access external data sources) (pg. 23, paragraph 350 – pg. 24, paragraph 352; pg. 24, paragraph 353-354).

As to claims 32 and 33, CHONG teaches a memory for storing data associated with a custom activity for access by a process (application) being executed by a

Art Unit: 2195

processor the memory comprising: a structure (workflow) specifying an input tag that obtains a value for an input hashtable to be used as an argument in a method (via the setvariableaction tag / modelinterface tag / modelvardef tag / databindinginterface tag / databinding tag wherein a data binding identifies both the input variables and the output variables), specifying an output tag that specify parameters that define what to do with parameters in an output hashtable including output arguments associated with the method (via performing any action tag / databindinginterface tag / databinding tag wherein a data binding identifies both the input variables and the output variables), and specifying design tags that define a user interface associated with the custom activity (listview tag / devicebasedview tag) (pg. 16, paragraph 284 – 297; pg. 24, paragraph 351-357).

As to claim 34, CHONG teaches a memory for storing data for access by a process (application) being executed by a processor, the memory comprising: a structure defining a custom activity (component / object / workflow step) implemented in a process management system (development system / workflow environment) by defining a package (component) (via componentization) (pg. 29, paragraph 431-434) for importing packages external to the process management system (via using the adapters to access external data sources) (pg. 23, paragraph 350 – pg. 24, paragraph 352; pg. 24, paragraph 353-354), defining an init method for defining initialization tasks associated with the custom activity (pre-action tasks on the component) (pg. 15, paragraph 267-271; pg. 12, paragraph 231-234), and defining a perform method for

executing tasks associated with the custom activity (action tasks on the component) (pg. 15, paragraph 267-271; pg. 12, paragraph 231-234).

As to claim 37, CHONG teaches a system for creating and implementing custom activities (components / objects / workflow steps) in a process management environment (development system / workflow environment), comprising: a processor; and a memory containing instructions executable by the processor (pg. 8, paragraph 177-178) to: receive a selection to add a custom palette (via componentization packaging the visual icon of a component to be loaded in a palette of another system such that the component is available to the other user) (pg. 29, paragraph 430-434) (pg. 14, paragraph 258); receiving information reflecting an identifier associated with the custom palette (via receiving the Jar file and using the component by dragging and dropping its visual icon); and assigning a visual representation (visual icon) to the custom palette reflecting a custom activity that may be used in an automated workflow process (via componentization packaging the visual icon of a component to be loaded in a palette of another system such that the component is available to the other user) (pg. 29, paragraph 430-434) (pg. 14, paragraph 258) to access information external to the process management environment (via using the adapters to access external data sources) (pg. 23, paragraph 350 – pg. 24, paragraph 352; pg. 24, paragraph 353-354).

As to claim 38, CHONG teaches a system for creation and implementing a custom activity (component / object / workflow step) in a process management

Art Unit: 2195

environment (development system / workflow environment) comprising: a processor; and a memory containing instructions executable by the processor (pg. 8, paragraph 177-178) to: receive a request to generate a palette associated with the custom activity; assign the custom activity to the palette (via performing componentization wherein the component is added to the IDE by copying icon files into a directory defined by the user such that components are added to the palette) (pg. 29-30, paragraph 433); and determine activation of the custom activity (component) based on a manipulation (drag and drop) associated with the palette pg. 29, paragraph 430-434) (pg. 14, paragraph 258), wherein the custom activity accesses resources external to the process management environment (via using the adapters to access external data sources) (pg. 23, paragraph 350 – pg. 24, paragraph 352; pg. 24, paragraph 353-354).

As to claim 39, CHONG teaches a system for creating an implementing a custom activity (component / object / workflow step) in a process management environment (development system / workflow environment), comprising: a processor; and a memory containing instructions executable by the processor (pg. 8, paragraph 177-178) to: receive a first file (adapter / class) defining with an interface with a package external to the process management system (external data source); receive a second file defining parameters (variables) that the first file uses; archive the first and second file in an archive file (via receiving the classes and variable information packaged in a Jar file) (via componentization) (pg. 29, paragraph 431-434); and invoke the first and second file based on a manipulation of an image reflecting the custom activity in a visual process

map (workflow diagram) reflecting an automated workflow process (via dragging and dropping the icon representing the component to a diagram of the workflow) (pg. 13, paragraph 257 – 258; pg. 14, paragraph 262).

As to claim 2, CHONG teaches the data representation file includes a section that determines the appearance of a representation reflecting the activity (visual icon) (pg. 29, paragraph 431-434).

As to claim 3, CHONG teaches the class file includes a method that is configured (via the adapter bindings and the functional black box interface) to obtain a value of a parameter defined in the data representation file (pg. 29, paragraph 431-434; pg. 24, paragraph 351-357).

As to claim 4, CHONG teaches receiving data that defines a package for the class file; and receiving data that defines methods that retrieve and set values to variables to be used by the activity (via componentizing the behavior of the component and instruction on how to assemble the resources of the component into the application (pg. 29, paragraph 434-434).

As to claim 5, CHONG teaches receiving data that reflects a method that defines variables that are constant across all instances of the activity (via the black-box

interface / data bindings that defines both input and output variables) (pg. 29, paragraph 430-434; pg. 24, paragraph 353-357; pg. 30, paragraph 433).

As to claim 6, CHONG teaches the method is associated with an input hashtable to define values of a variable used by the activity (variable to be passed in via the black-box interface / data bindings) (pg. 29, paragraph 430-434; pg. 24, paragraph 353-357; pg. 30, paragraph 433).

As to claim 7, CHONG teaches receiving data (for componentization or at the receiving node of the Jar file) reflecting a method that defines values for variables in a first hashtable and retrieves values for variables from a second hashtable (variable to be passed in and out via the black-box interface / data bindings) (pg. 29, paragraph 430-434; pg. 24, paragraph 353-357; pg. 30, paragraph 433).

As to claim 8, CHONG teaches receiving data reflecting a method that releases resources used by an application that implements the activity when the application is unloaded from the process management system (via the delete action) (pg. 14-15, paragraph 265).

As to claim 9, CHONG teaches receiving data reflecting a first section that defines a type and name of the class file (adapter); receiving data reflecting a second section that defines parameters with values that remain constant within all instances of

the activity (via the variables or data bindings provided); receiving data reflecting a third section that sets values for selected parameters within a first hashtable (via an action operation on the variables); receiving data reflecting a fourth section that defines what to do with parameters included in a second hashtable (via instruction on how to assemble the resources in the component into the application); and receiving data reflecting a fifth section associated with a visual representation associated with the activity (visual icon of the component) (pg. 29, paragraph 431-435).

As to claim 10, CHONG teaches packaging the first and second data into one of a JAR file or a ZIP file (pg. 29, paragraph 431).

As to claim 11, CHONG teaches locating the packaged data; and receiving data reflecting a visual representation (visual icon) that corresponds to the packaged files (pg. 29, paragraph 431-434).

As to claim 13, CHONG teaches the file is an archive file and includes the visual representation (pg. 29, paragraph 431-434).

As to claim 15, CHONG teaches the model is an image reflecting the custom activity (visual icon) (pg. 29, paragraph 431-434).

As to claim 16, CHONG teaches packaging the file and model into an archive file (pg. 29, paragraph 431-434).

As to claim 17, CHONG teaches associating the custom activity with a workflow process managed by the process management system (via dragging and dropping the component via its visual icon into the workflow diagram) (pg. 13, paragraph 257 – 258; pg. 14, paragraph 262).

As to claim 18, CHONG teaches determining a position of the model in a visual process map (workflow diagram) reflecting the workflow process; and invoking the custom activity in the workflow process based on the determination (via linking the components of the workflow diagram) (pg. 13, paragraph 257 – 258; pg. 14, paragraph 262).

As to claim 19, CHONG teaches the at least one file includes a Java class file (object oriented class) (pg. 7, paragraph 149; pg. 24, paragraph 351-352; pg. 25, paragraph 369) and an XML description file (pg. 8, paragraph 176; pg. 19, paragraph 325).

As to claim 22, CHONG teaches receiving package information associated with the first file (adapters) that implement packages external to the process management system (via using the adapters to access the external data sources and manipulate the

external data sources) (pg. 23, paragraph 350 – pg. 24, paragraph 352; pg. 24, paragraph 353-354).

As to claim 23, CHONG teaches receiving data that interacts with parameters associated with a hashtable defined in the second file (via performing functions to acquire the in variables for the adapter) (pg. 24, paragraph 353-357).

As to claim 24, CHONG teaches receiving data associated with the second file that defines at least one hashtable used by the first file (via defining or retrieving the in variables for the adapter) (pg. 24, paragraph 353-357).

As to claim 25, CHONG teaches the first file reflects a class file (object oriented class) (pg. 7, paragraph 149; pg. 24, paragraph 351-352; pg. 25, paragraph 369) and the second file reflects an XML file (pg. 8, paragraph 176; pg. 19, paragraph 325).

As to claim 26, CHONG teaches archiving the files in a n archive file consisting of one of a JAR file and a ZIP file (pg. 29, paragraph 431-434).

As to claims 28 and 29, CHONG teaches the data representation is an XML description file (pg. 8, paragraph 176; pg. 19, paragraph 325).

As to claim 35, CHONG teaches the perform method is associated with at least a first hashtable (variable) including values corresponding to data fields and a second hashtable including values to be placed in the data fields (pg. 24, paragraph 353-357).

As to claim 36, CHONG teaches the custom activity may have a plurality of instances and wherein the init method defines an association with resources external to the process management system and are shared by all instances of the custom activity (via a component is a container of other components and the components have instructions on how to assemble the resources in the component into the application using that component) (pg. 29, paragraph 431-434).

As to claims 40-65, reference is made to a computer readable medium that corresponds to the method of claims 1-26 and is therefore met by the rejection of claims 1-26 above.

As to claims 66-91, reference is made to a system that corresponds to the method of claims 1-26 and is therefore met by the rejection of claims 1-26 above.

Conclusion

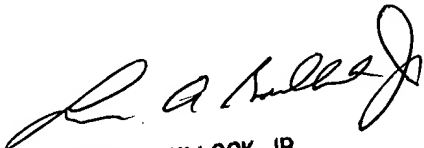
Any inquiry concerning this communication or earlier communications from the examiner should be directed to Lewis A. Bullock, Jr. whose telephone number is (571)

272-3759. The examiner can normally be reached on Monday-Friday, 8:30 a.m. - 5:00 p.m..

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Meng An can be reached on (571) 272-3756. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

January 7, 2006



LEWIS A. BULLOCK, JR.
PRIMARY EXAMINER